

PULSSINLEVEYSMODULAATION
KÄYTTÖ
MOOTTORIOHJAUKSESSA

LAHDEN AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Tietokone-elektroniikka
Opinnäytetyö
Kevät 2011
Kalle Lunkka

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

LUNKKA, KALLE: Pulssinleveysmodulaation käyttö moottorihjauksessa

Tietokone-elektroniikan opinnäytetyö, 29 sivua, 5 liitesivua

Kevät 2011

TIIVISTELMÄ

Tässä opinnäytetyössä käsitellään pulssinleveysmodulaation käyttöä moottorihjauksessa. Tarkastellaan sähkömoottoreita sekä servotekniikkaa ja niiden toimintaperiaatteita sekä sitä miten niitä PWM-tekniikalla ohjataan. Työstä tehdään myös käytännön sovellus, jossa ohjataan mikro-ohjaimen avulla tasavirtamoottoria sekä servomoottoria.

Tämän työn tavoitteena on perehtyä pulssinleveysmodulaation teoriaan sekä toimintaperiaatteeseen. Tutkitaan myös, minkä vuoksi se on kehitetty ja yleistynyt elektroniikkakäytössä. Vertaillaan sitä muihin vastaaviin tekniikoihin ja pohditaan sen heikkoja sekä vahvoja puolia.

Työssä kerrotaan myös DC-sekä servomoottoreiden ominaisuuksista ja toimintaperiaatteesta. Työssä käytetään Atmelin AT90USB1287-mikro-ohjainta ja perehdytään sen ominaisuuksiin niiltä osin kuin niitä työssä hyödynnetään. Käytännön sovellukseksi tehdään linjanseurantarobotti, jossa työntövoimana käytetään DC-moottoria, ohjaukseen servomoottoria, linjan seurantaan kontrasteja havaitsevia QRD1114-antureita sekä datan käsittelyyn ja moottoreiden ohjaukseen AT90USB1287-mikro-ohjainta.

Avainsanat: pulssinleveysmodulaatio, tasavirtamoottori, servomoottori, mikrokontrolleri, mikro-ohjain

Lahti University of Applied Sciences
Degree Programme in Information Technology

LUNKKA, KALLE: Using pulse width modulation for motor control

Bachelor's Thesis in Computer Electronics

29 pages, 5 appendices

Spring 2011

ABSTRACT

The subject of this thesis is the use of pulse width modulation, PWM, for motor control. Electric motors and servo technology were examined, along with their operating principles and how they are controlled with the PWM technique. A practical application was also created to demonstrate the PWM technique in action.

At the beginning of the thesis, the theory and operating principle of pulse width modulation were studied. Research was done to determine the reasons for the development of pulse width modulation and why it is so widely used in embedded electronics. The PWM technology was compared with other similar technologies, and its weaknesses and strengths were analyzed.

A line tracking robot was created as the practical application. It uses a DC motor for propulsion and a servo motor for steering. For line tracking the robot uses QRD1114 sensors that are able to sense a difference in contrasts. For the data processing and motor control there is Atmel's AT90USB1287 microcontroller. Its properties were also examined to the extent that they were used in the application.

Key words: Pulse width modulation, microcontroller, DC motor, servo motor

SISÄLLYS

1	JOHDANTO	1
2	PULSSINLEVEYSMODULAATIO	2
2.1	PWM-tekniikan edut ja haitat	3
2.2	PWM-tekniikan käyttö sovelluksissa	5
3	SÄHKÖMOOTTORIT	6
3.1	Tasavirtamoottori	6
3.2	Servotekniikka	8
3.2.1	Anturit	9
3.2.2	Vahvistin	10
4	TYÖN TOTEUTUS	12
4.1	Mikro-ohjain AT90USB1287	13
4.1.1	PWM-kanavan-0A konfigurointi	13
4.1.2	PWM-kanavan-1A konfigurointi	14
4.2	USB-ohjelmointiliitin	15
4.3	Jännitteen regulointi	17
4.4	L293D-ohjainpiiri	18
4.5	IR sensori QRD1114	19
4.6	Prototyypin rakentaminen	21
5	OHJELMAKOODIN SUUNNITTELU	24
6	TYÖN ANALYSOINTI	26
	LÄHTEET	28
	LIITTEET	30

1 JOHDANTO

Nykyaikaisessa tekniikassa on usein käytössä pieniä sähkömoottoreita, esimerkiksi uudessa autossa voi olla kymmeniä pieniä sähkömoottoreita. Näitä löytyy auton ikkunoista, pyyhkijöistä sekä jopa ajojärjestelmistä. Jokaista näistä moottoreista täytyy ohjata, ja hyvin usein ohjaus tapahtuu pulssinleveysmodulaation (pulse width modulation, PWM) avulla.

Tässä opinnäytetyössä tutustutaan ensin pulssinleveysmodulaation toimintaperiaatteeseen ja sen hyötyihin sekä ongelmiin sen käytössä. Tämän jälkeen kerron erilaisista sähkömoottoreista ja niiden tekniikasta sekä siitä, kuinka sähkömoottoria voidaan ohjata servojärjestelmän avulla käyttäen PWM-signaalia.

Näitä tekniikoita demonstroidakseni rakensin linjanseurantarobotin, jonka tarkoituksena on seurata tummaa linjaa valkoista pohjaa vasten. Robotti käyttää työntövoimanaan tasavirtamoottoria, jonka nopeutta säädellään PWM-signaalin avulla. Ohjaukseen käytetään tasavirtaservomoottoria, jonka ohjaukseen käytetään myös PWM-signaalia, mutta servon ohjauksessa täytyy ottaa huomioon eri asioita kuin tavallisen tasavirtamoottorin.

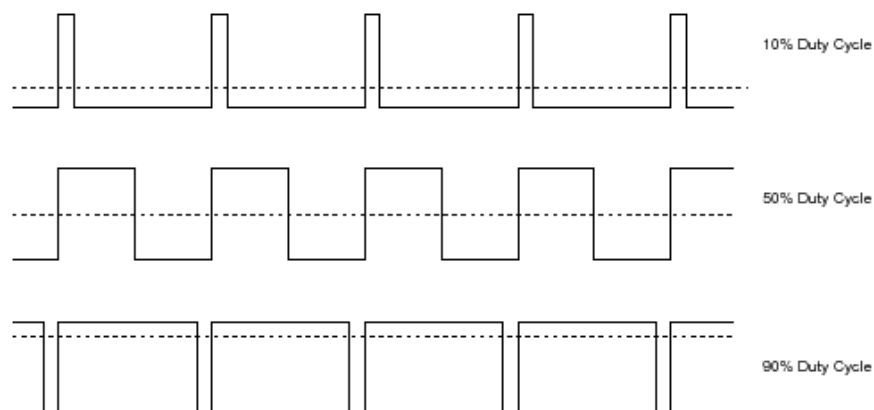
2 PULSSINLEVEYSMODULAATIO

Pulssinleveysmodulaatio, englanniksi pulse width modulation (PWM), on tekniikka, jolla rajoitetaan syötetyn virran määrää elektroniselle laitteelle, kuten sähkömoottorille. Virtaa rajoitetaan katkaisemalla virta ja kytkemällä se takaisin hyvin nopealla tahdilla, riippuen sovelluksesta jopa kymmeniätuhansia kertoja sekunnissa. Näiden on- ja off-jaksojen suhde määrää lopullisen virran, joka pääsee laitteelle. Tätä kutsutaan duty cycleksi (pulssisuhde), joka ilmoitetaan prosentuaalisena lukuna, esimerkiksi duty cycle 100 % tarkoittaa virran pääsevän rajoittamattomana laitteelle.

$$D = \frac{t}{T}$$

Kaava 1. Kaava duty cyclen laskentaan

Kaavan 1 merkinnät tarkoittavat: D = Duty cycle, t = on-jakson pituus, T = on- sekä off-jaksojen yhteenlaskettu aika. Kuviossa 1 havainnollistetaan PWM-signaalin duty cycleä.



KUVIO 1. Esimerkkejä PWM-signaalista (PWM PIC : Pulse Width Modulation 2011)

Yksinkertaisimmillaan PWM-signaalin voi toteuttaa kiteellä, joka tuottaa sahalaita- tai kolmioaaltoa, ja syöttämällä tämä sekä vertailujännite komparaattoriin, jolloin tuloksi saadaan kanttiaalto.

PWM-signaalin voi tuottaa myös useimmilla mikrokontrollereilla, joissain on jopa tätä tarkoitusta varten suunnitellut PWM-kanavat. Signaalin luomiseksi käytetään laskuria, jota kasvatetaan haluttuun lukuarvoon saakka ja jonka jälkeen se nollataan ja aloitetaan alusta. Kun laskurin arvo on suurempi kuin vertailuarvo, PWM-pinnin tulo muutetaan, esimerkiksi nolasta ykköseksi tai päinvastoin. (Bräunl 2008.)

2.1 PWM-tekniikan edut ja haitat

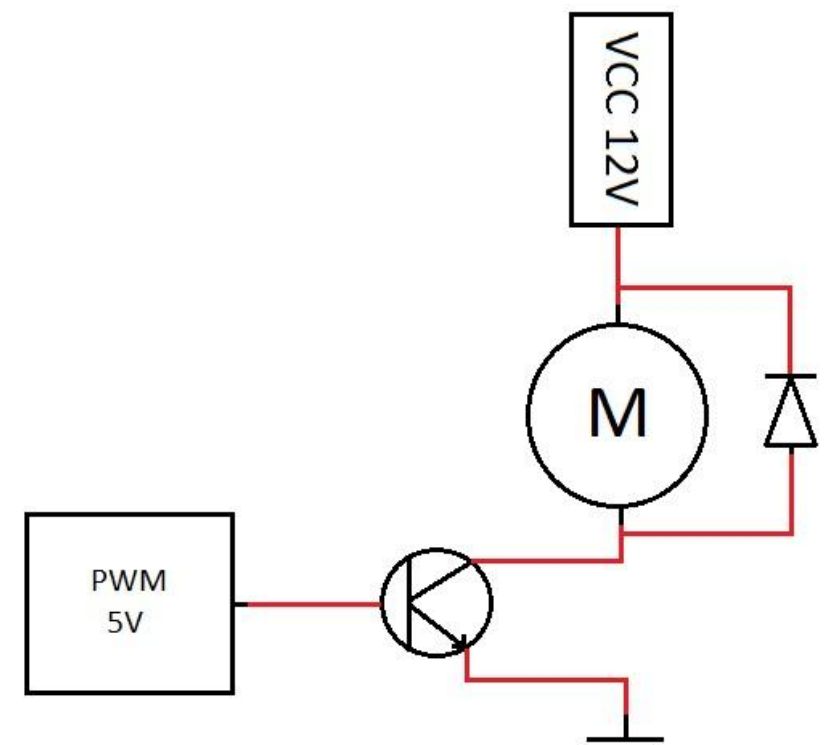
PWM-tekniikalla on muihin käytössä oleviin tehon muunnostekniikoihin verrattuna useita vahvuuksia, mutta myös joitakin ongelmia, jotka on hyvä ottaa huomioon suunnitellessa laitteita, jotka hyödyntävät sitä. PWM-tekniikka onkin nykyään hyvin yleisesti käytetty erilaisissa sulautettujen järjestelmien sovelluksissa.

Yksi näistä vahvuuksista on pulssinleveysmodulaatiolla saavutettava erinomainen hyötysuhde. Tämä johtuu siitä, että off-jakson aikana virtaa ei piirissä kulje lähes ollenkaan, ja tällöin tehohäviö jää hyvin pieneksi. Muita tekniikoita, kuten jännitteen jakoa tai säädettävää vastusta, käyttäessä suuri osa tehosta muuttuu hukkalämmöksi. (Netrino.com 2001.)

Toinen PWM-tekniikalla saavutettava etu on sen digitaalisuus. Koska PWM-signaali on yhdellä ajan hetkellä joko täysin maatasossa tai käyttöjännitteen suuruinen, ei sitä tarvitse muuntaa analogiseksi matkalla digitaalisesti ohjattavalle laitteelle. Digitaaliseen signaaliin ei myöskään vaikuta pieni kohina lainkaan, sillä kohinan tulee ylittää vertailujännite muuttaakseen signaalin tilan virheelliseksi. (Netrino.com 2001.)

Toisinaan varsinkin suuria jännitteitä käytettäessä saattaa PWM-signaalin käytöstä koitua haittaääniä. Nämä saadaan poistettua kuitenkin nostamalla PWM-signaalin taajuus 20 kHz, jolloin ihmiskorva ei enää kuule niitä. (Embeded robotics 2008.)

Nykyään PWM-signaalia luodaan usein mikro-ohjaimilla, jotka käyttävät pieniä käyttöjännitteitä. Signaalilla voidaan kuitenkin haluta ohjata moottoria, jonka käyttöjännite voi olla kymmeniä tai jopa satoja voltteja, ja tällöin on edessä ongelma. Tämän ongelman ratkaisuun voi käyttää joko FET- (Field-Effect Transistor) tai muuta vastaavaa transistoria. Kun kytketään kontrollerilta tuleva PWM-signaali transistorin basenapaan, lähdejännite kollektorille ja kuorma emitterille, toimii transistori tällöin kuin valokatkaisija, joka kytkeytyy päälle ja pois PWM-signaalin tahdissa. Kuviossa 2 olen kuvannut tätä kytkentää. Siinä näkyy myös diodi moottorin kanssa rinnan kytkettynä. Tämä diodi estää moottorin magneettikentän tuottamaa vastakkaista jännitettä vahingoittamasta transistoria. (Eleinmec.com 2007.)



KUVIO 2. Transistorikytkentä

2.2 PWM-tekniikan käyttö sovelluksissa

PWM-tekniikka on nykyään laajalti käytetty sulautetuissa järjestelmissä. Koska nykyään sähköä kuluttavia laitteita on paljon ja sähkön hinta on korkea, ei sitä haluta tuhlaa. PWM-signaali on helppo toteuttaa edullisillakin komponenteilla. Tästä syystä PWM-tekniikkaa käytetään lähes aina siellä missä ennen käytettiin esimerkiksi potentiometriä. Hyvä esimerkki on monista tietokoneista löytyvät tuulettimen nopeuden säätimet. Säätimessä on sekä potentiometri että muu elektronikka PWM-signaalin luontia varten. Vaikka tuulettimen nopeuden säätö voitaisiin toteuttaa suoraan potentiometriä kääntämällä, on se toteutettu PWM-signaalilla, jota vain ohjataan potentiometriä kääntämällä. Näin säästetään potentiometrin muutoin lämmöksi muuttama energia.

3 SÄHKÖMOOTTORIT

Maailmassa on käytössä lukuisia sähkömoottoreita. On arvioitu, että sähkömoottoreissa käytetään kaikkiaan puolet kaikesta tuotetusta sähköstä. Koska sähköenergia on kallista, sähkömoottoreista kannattaa tehdä mahdollisimman laadukkaita ja energiatehokkaita. Pienimmillä moottoreilla hyötysuhde on huono, mutta suurilla sähkömoottoreilla hyötysuhde vaihtelee 95:n ja 98:n prosentin välillä. Yleisesti ottaen moottoreiden, joita käytetään teollisuudessa, hyötysuhde on yli 80 %. (Lappeenranta university of technology 2011.)

Sähkömoottoreita esiintyy paljon tehtaissa, mutta suurin osa niistä löytyy kuluttajien laitteista. Liikenteen ja liikkuvien työkoneneiden muuttuessa tulevaisuudessa sähköisiksi, sähkömoottoreiden käyttö kasvaa entisestään. Sähköauto käyttää useimmin työntövoimanaan joko nopeasti käyvää oikosulkumoottoria tai kesto- magneettimoottoria, joka voi olla rakennettuna esimerkiksi pyörännapaan. (Lappeenranta university of technology 2011.)

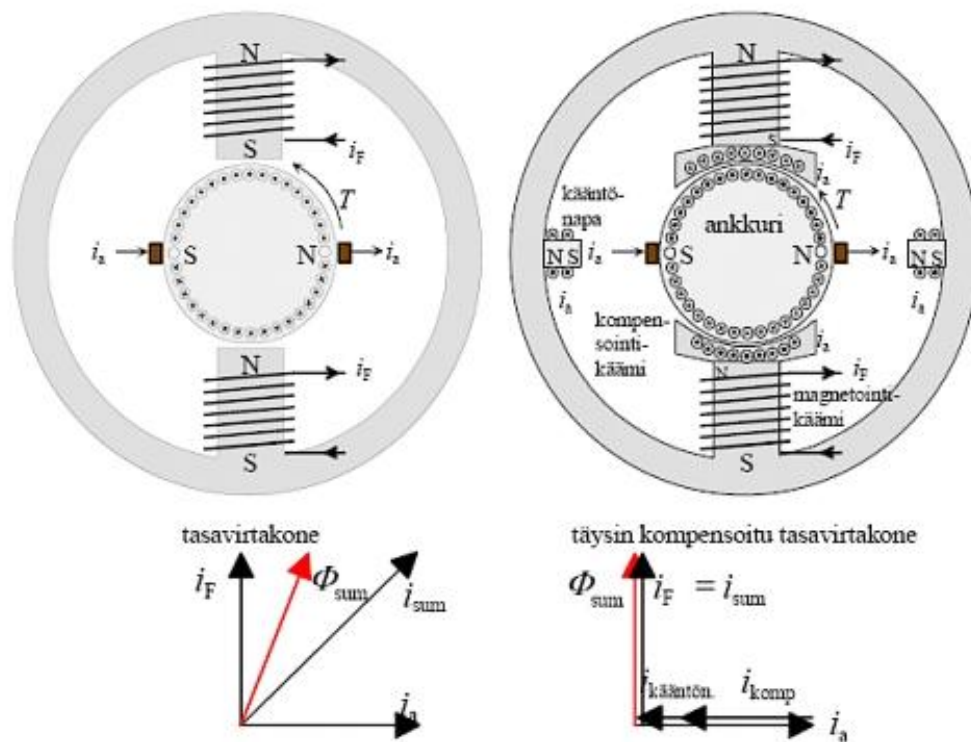
3.1 Tasavirtamoottori

Ensimmäiset säädettävät sähkömoottorit tehtiin tasavirtamoottoreista. Erityisen hyväksi tähän tarkoitukseen osoittautui niin kutsuttu vierasmagnetoitu täysin kompensoitu tasavirtamoottori, sillä siihen on helppo toteuttaa yksinkertaisilla välineillä pyörimisnopeuden- sekä väännönsäätö. Tasavirtamoottori onkin ollut vallitsevassa asemassa sähkömoottoreiden maailmassa lähes koko 1900-luvun. Oikosulkumoottorin yleistyttyä teollisuuden käytössä 1970-luvun alussa tasavirtamoottorin suosio laski. Teollisuudessa tasavirtamoottorin huolto tuli kalliiksi, sillä se vaati säännöllistä hiili- ja kommutaattorihuoltoa, ja lisäksi se ei kestä kovin suuria pyörimisnopeuksia moottorin kuparilangasta valmistetun urakäämityksen vuoksi. Myös tasavirtamoottorin heikompi hyötysuhde, verrattuna oikosulkukoneen hyötysuhteeseen, vaikutti ratkaisevasti tasavirtamoottorin käytön vähentymiseen teollisuudessa. (Lappeenranta university of technology 2011.)

Tasavirtamoottoreita käytetään edelleen pienissä määrin, vaikka niiden suosio onkin laskenut. Tasavirtakoneet vaikuttivat omalta osaltaan myös liikennevälinekäyttöjen kehitykseen. Niin kutsuttuja tasavirtasarjamoottoreita käytettiin varhaisten sähkökäyttöisten linja-autojen varustelussa. Tasavirtasarjakoneiden etuna on, että niitä pystyy käyttämään ilman tehoelektronista säätöä. Esimerksi VR:n SR1-sähkövetureissa on käytetty tasavirtakonetekniikkaa tyristorisiltoineen nykypäivänäkin, mutta uudemmissa SR2-vetureissa on käytetty eri varusteluja. SR2-vetureita on varustettu oikosulkumoottorein ja taajuudenmuuttajin. (Lappeenranta university of technology 2011.)

Tasavirtamoottorin hyötysuhdetta on mahdollista parantaa lisäämällä moottorin staattorin napoihin niin kutsutut kompensointikäämit ja magneettipiirin kehällä hiiliharjojen kohdalle kääntönavat. Näissä käämeissä kierrätetään ankkurivirtaa ja näin ollen voidaan estää ankkurireaktion vaikutus. Täysin kompensoidulla tasavirtamoottorilla on hyvä hyötysuhde ja sitä on helppo ohjata. Näistä syistä se on ollut jo pitkään käytetyin sähkömoottori tarkkuutta ja ohjauskykyä vaativiin sovelluksiin. Täysin kompensoidun tasavirtamoottorin kallis hinta sekä hiiliharjojen ja kommutaattorin huollon tarve kuitenkin kannustaa etsimään parempia ratkaisuja teollisuuden käyttöön. Kuviossa 3 nähdään tasavirtamoottorin sekä täysinkompensoidun tasavirta moottorin rakenteelliset erot. (Lappeenranta university of technology 2011.)

Ratkaisuksi näyttää nousseen mikro-ohjaimien sekä taajuuden muuttajan myötä paremmat mahdollisuudet ohjata oikosulkumoottoreita. Oikosulkumoottori (AC- eli vaihtovirtamoottori) on noussut viime vuosikymmeninä tasavirtamoottorin ohitse teollisuuden käytetyimmäksi moottoriksi. Se on rakenteeltaan yksinkertainen, kestävä sekä hinnaltaan tasavirtamoottoria halvempi. (Lappeenranta university of technology 2011.)



KUVIO 3. Tasavirtamoottori, täysin kompensoitu tasavirtamoottori sekä niitä kuvaavat virtavektorit (Lappeenranta university of technology 2011)

3.2 Servotekniikka

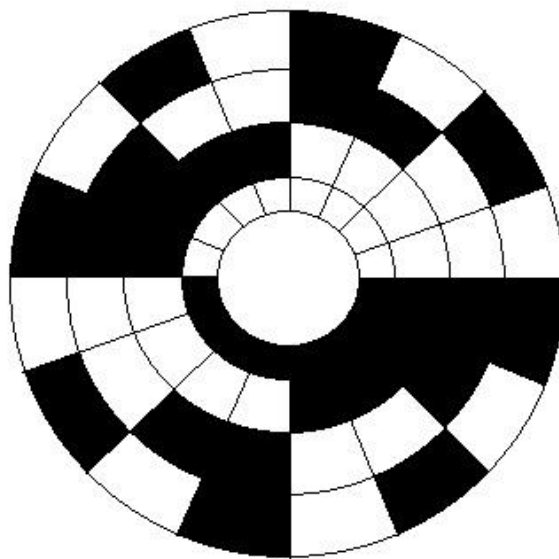
Servomoottori on joko tasavirta- tai vaihtovirtamoottori, jonka pyörimistä voidaan tarkasti säädellä takaisinkytkennän avulla. Tasavirtaservomoottorille on tyypillistä, että joko staattorin tai roottorin käämit voidaan korvata kestopagneeteilla, ja niiden pyörimisnopeutta ja vääntömomenttia voidaan säätää laajalti. (Johnson & Kördel 2003, 22 – 24.)

Servomoottoriin on usein hyödyllistä saada hitaampi pyörimisnopeus ja kasvattaa vääntömomenttia. Tämä saadaan aikaiseksi hammaspyörillä tai hihnoilla. Momentin lisäys on kääntäen verrannollinen välityssuhteeseen, pois lukien vaihteiston hävikki. Vaikka servojärjestelmän nopeuden laskemiselta tai vääntömomentin nostamiselta harvoin välttyy, olisi sitä syytä välttää, sillä se nostaa tuotantokustannuksia. (Johnson & Kördel 2003, 24 – 26.)

3.2.1 Anturit

Säädettäessä moottorin asentoa täytyy sen liikkeistä saada tietoa. Tämä tieto kerätään antureiden avulla. Servojärjestelmistä löytyy aina yksi tai useampi tällainen anturi. Antureilla mitataan asentoa, nopeutta, virran kulkua moottorin lävitse sekä moottorin tuottamaa vääntömomenttia. Jokaista suuretta varten löytyy erilaisia mittaustapoja. (Johnson & Kördel 2003, 39 – 41.)

Moottorin asentoa voidaan mitata yksinkertaisesti potentiometrillä, joka on kiinnitettyä moottorin varteen. Moottorin pyöriessä potentiometri kiertyy ja vastus muuttuu. Asentoa voidaan myös tarkkailla moottorin varteen sijoitetun koodikiekkon eli ns. absoluuttianturin avulla. Kiekossa on valoa läpäiseviä ja läpäisemättömiä kohtia. Näistä muodostuu sektoreita, joista jokainen on erilainen. Kun kiekkoa valaistaan toiselta puolelta hohtodiodeilla ja toisella puolella tarkkaillaan läpituloavaa valoa fotodiodejen avulla, saadaan tieto siitä missä asennossa moottori kulloinkin on. Kuviossa 4 on kuvattu nelikanavaisen absoluuttianturin koodikiekkon. Lisäämällä kanavia voidaan tarkkuuden resoluutiota kasvattaa huomattavasti. Neljällä kanavalla saadaan vain $360^\circ / 16 = 22,5^\circ$ tarkkuus. (Johnson & Kördel 2003, 39 – 41.)



KUVIO 4. Absoluuttianturin binäärikiekkon jossa jokainen sektori on erilainen

Koodikiekon avulla voidaan mitata myös moottorin pyörimisnopeutta. Tällöin mitataan diodejen ohitse kulkevien aukkojen lukumäärää tietyn ajan kuluessa. Nopeuden mittaamiseen riittää yksi kanava jolloin koodikiekossa tulisi kuitenkin olla aukkoja tiheästi tarkkuuden säilyttämiseksi. (Johnson & Kördel 2003, 39 – 41.)

Vääntömomenttia voidaan mitata esimerkiksi venymäliuska-anturilla. Venymäliuska-anturi koostuu paperi- tai muovikalvon sisään silmukoiksi sijoitetusta ohuesta metallilangasta. Lankaan kohdistuessa venytystä metallin kimmoisalla alueella, resistanssi muuttuu suoraan verrannollisesti venymään. (Johnson & Kördel 2003, 47.)

3.2.2 Vahvistin

Servovahvistimesta löytyy operaatiovahvistimia, tehovahvistimia sekä uusia A/D-muuntimia analogisten sekä digitaalisten komponenttien yhteen liittämistä varten. Servovahvistimen tehtävä on tarkkailla antureilta tulevaa dataa sekä käyttäjältä tulevaa ohjaussignaalia. Näiden avulla vahvistin ohjaa moottoria parhaansa mukaan. (Johnson & Kördel 2003, 53 – 56.)

Servovahvistimet ovat kehittyneet paljon 1960-luvulta, jolloin ensimmäiset vahvistimet tulivat markkinoille. Kehityksestä saa suurelta osin kiittää mikroprosessorien- sekä puolijohdetekniikan nopeaa kehitystä. Nykyaikaisen servojärjestelmän tulisi täyttää tietyt laatuksiteerit, joita ovat nopea vasteaika ja kyky seurata nopeasti tulosignaalin muutoksia, vakaa toiminta, jottei vahvistin joudu itsevärähtelytilaan, kyky seurata lähtösignaalia tarkasti sekä hyvä kohinan sietokyky. (Johnson & Kördel 2003, 69 – 71.)

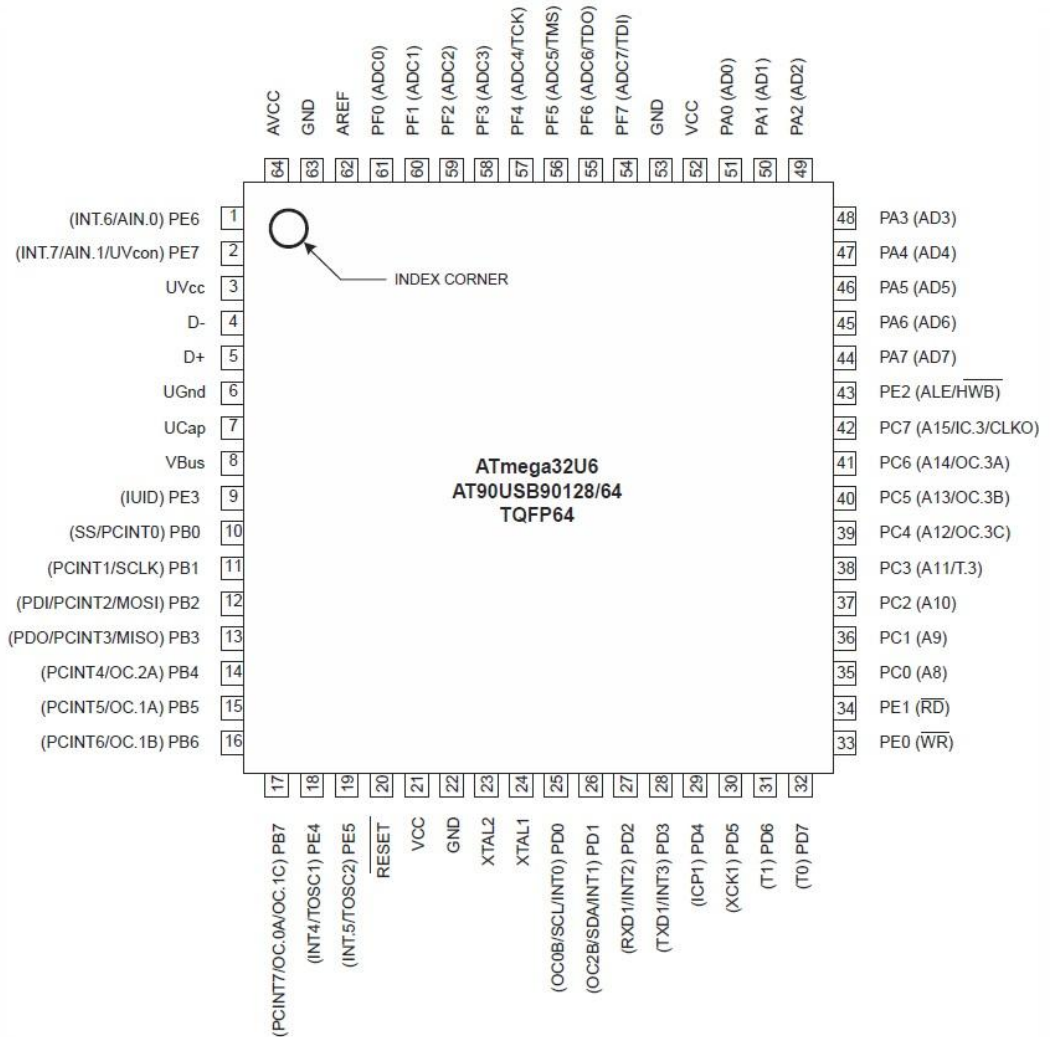
Servojärjestelmään saattaa kohdistua vaikutuksia järjestelmän ulkopuolelta. Moottorin kuorman kasvaessa on järjestelmän pystyttävä reagoimaan, jotta nopeus saadaan pidettynä vakiona. Tätä reagoointia kutsutaan säädöksi. Säättöön on käytössä kolme parametriä. P-toiminto, joka tarkoittaa tulosignaalin muutokseen verrannollista säätötoimintoa. Tämä ohjaa vahvistuksen suuruutta. Vahvistuksen lisääminen parantaa järjestelmän tarkkuutta sekä nopeutta, mutta liiallinen vahvistus voi kui-

tenkin saada aikaan epävakauden. P-toiminnolla ei ole mahdollista poistaa poikkeamaa täysin. Tätä varten täytyy lisätä integroiva toiminto, jolloin saadaan aikaan PI-toiminto. PI-toiminnolla saadaan minimoitua lähtö- ja tulosignaaleiden poikkeamaa. Servojärjestelmän toimintaa voidaan parantaa vielä entisestään lisäämällä derivoiva toiminto, jolloin saadaan PID-toiminto. Derivointi parantaa järjestelmän reagointinopeutta. PID-toiminto on servojärjestelmissä yleisesti käytetty parametri. (Johnson & Kördel 2003, 60 – 71.)

Servovahvistimen toimintaa kuvaa hyvin tilanne, jossa operaatiovahvistimen toista tulosignaalia säädellään potentiometrin 1 avulla, jota säädetään käsin, ja toista tulosignaalia potentiometrin 2 avulla, joka kääntyy moottorin pyöriessä. Jos molempien potentiometrien vastus on yhtä suuri, on tällöin operaatiovahvistimen lähtö 0 eikä moottori liiku. Kun muutetaan potentiometrin 1 arvoa pienemmäksi, muuttuu vahvistimen kääntävän tulon jännite ei-kääntävään tuloon nähden positiiviseksi. Näin ollen vahvistimen tulo muuttuu negatiiviseksi ja moottori alkaa pyöriä kääntäen samalla potentiometriä 2. Potentiometrien ollessa taas samassa asennossa moottori pysähtyy. Vastaavasti muuttamalla potentiometrin 1 arvoa suuremmaksi kuin potentiometrin 2, muuttuu vahvistimen lähtö positiiviseksi ja moottori pyörii vastakkaiseen suuntaan. Tämä kytkentä toimii tosin vain kaksisuuntaisella tasavirtamoottorilla. (Johnson & Kördel 2003, 69 – 71.)

4 TYÖN TOTEUTUS

Tein käytännön sovellukseksi linjanseurantarobotin, joka seuraa neljän kontrasteja havaitsevan IR-sensorin avulla maassa olevaa mustaa linjaa. Ohjauksesta vastaa servomoottori ja työntövoimasta tavallinen tasavirtamoottori. Molempia moottoreita ohjataan PWM-signaaleilla, joita luodaan mikro-ohjaimen laskureiden avulla. Työn toteutusta pohtiessa halusin käyttää juuri näitä kahta moottorityyppiä, sillä tasavirtamoottorin sekä servon ohjaus PWM-signaalilla vaatii erilaisten asioiden huomioon ottamista. Lisäsin laitteeseen myös ohjelmointia varten USB-liittimen, jonka avulla laitteen ohjelmointi ja debuggaus on helppoa.



KUVIO 5. Atmelin AT90USB1287 mikro-ohjaimen pinnijärjestys (Atmel 2009)

4.1 Mikro-ohjain AT90USB1287

Mikro-ohjainta valittaessa Atmelin AT90USB1287 oli minulle luonnollinen valinta, sillä olin jo aikaisemmin tutustunut sen käyttöön sekä ominaisuuksiin.

AT90USB1287 on muiden Atmelin mikro-ohjainten tapaan 8-bittinen RISC- (Reduced instruction set) arkkitehtuuriin perustuva mikro-ohjain. Muita työni kannalta merkittäviä ominaisuuksia AT90USB1287 mikro-ohjaimessa ovat

- 16 MIPS (million instructions per second) suorituskky
- neljä 8-bittistä PWM-kanavaa
- kuusi PWM-kanavaa, joissa on ohjelmoitava 2–16-bittinen resoluutio
- 128 kilotavua flash-ohjelmamuistia
- USB yhteensopivuus
- hyvät virransäästöominaisuudet. (Atmel 2009.)

4.1.1 PWM-kanavan-0A konfigurointi

Sovellustani varten tarvitsin kaksi PWM-kanavaa. Työntövoiman antavaa tasavirtamooottoria ohjaamaan valitsin 8-bittisen laskurin Timer0, joka ohjaa portin B7 pinnissä olevaa PWM-kanavaa OC0A. Pinnijärjestys on esitetty kuviossa 5, josta PB7 löytyy pinnistä 17. PWM-kanava konfiguroidaan asettamalla ajastin/laskuri rekisterien TCCR0A sekä TCCR0B bitit haluttuun järjestykseen. Valitsin Fast-PWM-tilan, jossa laskuri laskee välillä 0 ja 255 ja vertaa sitä ohjelmoijan asettamaan 8-bittiseen OCR0A-arvoon. Kun arvot kohtaavat, asetetaan OC0A-kanava ykköseksi. Päästessään arvoon 255, asetetaan kanava OC0A takaisin nollatilaan, jonka jälkeen aloitetaan laskenta alusta. Tällä tavoin voidaan OCR0A-arvoa muuttamalla määritellä PWM-signaalin duty cycle. Jos esimerkiksi OCR0A-arvo on 128, on duty cycle 50 %. (Atmel 2009.)

4.1.2 PWM-kanavan-1A konfigurointi

Servomoottorin ohjaukseen Timer0-laskuria ei voi käyttää, koska pelkkä duty cyclen säätely ei riitä. Tarvitaan myös määrätty taajuus. Käyttämäni Paralaxin standard servon taajuus oli 20 ms eli 50 Hz. Taajuuden lisäksi tulee moottorista selvittää ohjauspulssin pituus. Paralaxin tapauksessa ilmoitettiin vain pulssin pituus (1,5 ms), jolla moottori on keskiasennossa. 1,5 ms vastaa taajuuden ollessa 20 ms 7,5 % duty cycleä. (Paralax Inc 2010.)

Mikro-ohjaimesta löytyy kuitenkin kaksi laskuria, joissa on säädettävä resoluutio: laskurit Timer1 sekä Timer3, joista valitsin Timer1:n. Samaan tapaan kuin konfiguroitaessa Timer0, myös Timerille1 löytyy TCCR1A- sekä TCCR1B-rekisterit, jotka asettamalla valitaan toimintatapa, jota halutaan käyttää. Timerille 1 löytyy kaksi Fast PWM -tilaa, joissa voidaan määritellä pulssintaajuus. Nämä ovat toimintatapa 14 sekä 15. Datalehdessä suositeltiin toimintatapaa 15, jos taajuutta olisi tarvetta muuttaa ohjelman ajon aikana. Tämä kuitenkin vaatisi käyttöönsä kaksi laskuriin 1 liitetyistä kanavista. En myöskään nähnyt tarvetta taajuuden muuttamiselle ajon aikana, sillä servomoottorini ohjaamiseen tarvitsin pelkästään 50 Hz:n taajuutta. Näin ollen valitsin toimintatavan 14, jossa laskuri laskee 16-bittiseen ICR1-rekisteriin talletettuun arvoon saakka, jonka jälkeen se nollaantuu samaan tapaan kuin Timer0. (Atmel 2009.)

Jotta saadaan PWM-kanavasta ulos oikea taajuus, täytyy ottaa huomioon muutamia seikkoja. Ensimmäiseksi täytyy tarkistaa mikro-ohjaimen kellotaajuus. AT90USB1287:n maksimikellotaajuus on 16 MHz, sovellukselleni ei näin suuresta kellotaajuudesta ole kuitenkaan suurta hyötyä, joten päätin käyttää 1 MHz:n taajuutta virran säästämiseksi. Varsinkin suurien kellotaajuuksien kohdalla on myös suositeltavaa käyttää laskurissa olevaa esijakajaa. Esijakaja konfiguroidaan TCCR1B-rekisteriä muokkaamalla. Esijakaja jakaa prosessorin taajuuden 8:lla, 64:llä, 256:lla tai 1024:llä, eli käytännössä laskuri kasvaa vain joka 8–1024 kerta riippuen käytetystä esijakajasta. Kun prosessorin kellotaajuus sekä käytettävä esijakaja ovat selvillä, voidaan laskea tarvittava ICR1-arvo kaavan 2 osoittamalla tavalla. (Atmel 2009.)

$$ICR1 = \frac{F_{cpu}}{F_{pwm} * N} = \frac{1MHz}{50 * 64} = 312$$

Kaava 2. ICR1-arvon laskentakaava

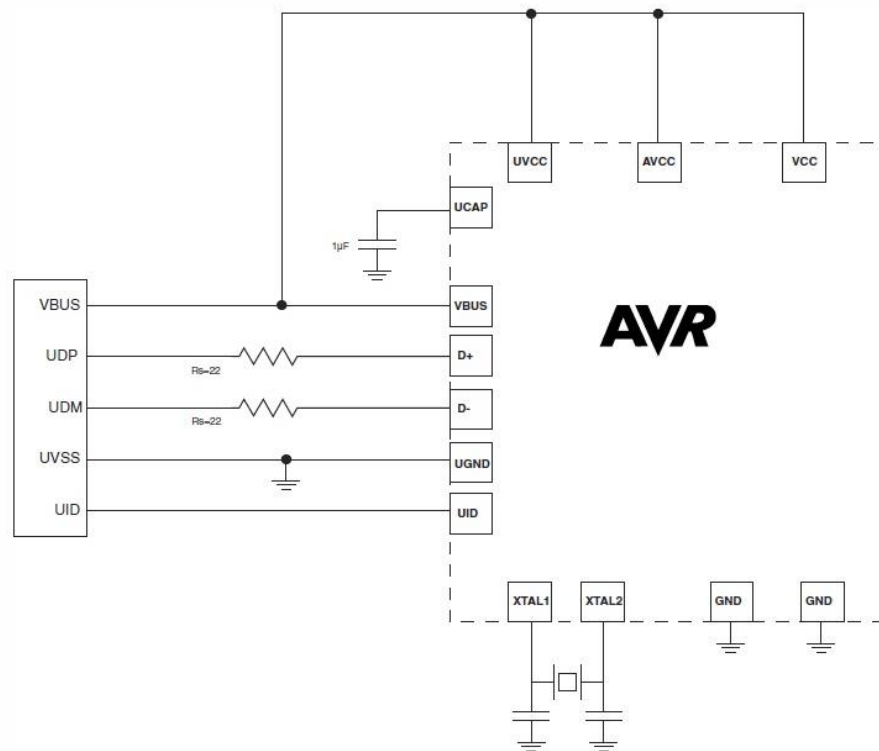
Kaavassa esiintyvät kirjainyhdistelmät ovat F_{cpu} = prosessorin taajuus, F_{pwm} = haluttu PWM signaalin pulssitaajuus ja N = esijakajan suuruus. Duty cycle määritellään samalla tavalla kuin timer0 tapauksessa, ainoana erona on, että myös vertailurekisteri OCR1A on 16-bittinen. Esimerkiksi 7,5 % duty cycle saadaan asettamalla OCR1A rekisteriin arvo 23, joka on siis 7,5 % ICR1 arvosta 312. (Extreme electronics 2007.)

4.2 USB-ohjelmointiliitin

AT90USB1287-mikro-ohjaimen pystyy helposti ohjelmoimaan Atmelin Flip (Flexible in-system Programmer) -ohjelmistolla. Atmelin datalehdessä löytyy ohjeet USB ominaisuuden käyttöönottoa varten. Valmiita kytkentäkaavioita on luotu erilaisille laitetyppeille sekä jännitteille. Valitsin liitännästä virran saavan device-liitännän. Tämä mahdollisti rakennusvaiheessa paremmin laitteiston testauksen, sillä sain käyttööni USB-liitännän kautta 5 voltin reguloidun jännitteen ilman ulkoista virtalähdettä, eikä kytkennästä ole haittaa laitteen toiminnan aikana, sillä se ei koskaan ole toiminnassa ulkoisella virralla sekä USB-liitännässä yhtä aikaa. (Atmel 2009.)

Toimiakseen luotettavasti USB-yhteys tarvitsee ulkoisen kellon, joka voi vaihdella 2 ja 16 MHz:n välillä. Mikro-ohjaimen sisäisesti tästä matalammasta taajuudesta muunnetaan ensin 48 MHz:n taajuuteen. Tästä 48 MHz:n taajuudesta se laskeaan 12 MHz:n taajuuteen täysinopeuksista USB-yhteyttä varten. Kellokiteen lisäksi USB-liitäntää varten tarvitaan kaksi 22 Ω :n vastusta datalinjoihin kaikkumisen ehkäisemiseksi, 1 μ F:n kondensaattori U_{cap} -liitäntään sekä 10 μ F:n suotokondensaattori käyttäjännitteen ja maalinjan väliin, joka datalehden mukaan on suositeltava, mutta jota ei ole kuvioon 6 lisätty. Jotta mikro-ohjaimen saa tilaan, jossa siihen voi ladata uutta ohjelmakoodia, täytyy pinni 43 HWB (Hardware

boot) sekä pinni 20 reset kytkeä. Suunnittelin molempia pinnejä varten oikosulkupala-kytkennän, sillä en osia tilatessa huomannut ottaa huomioon näiden nappien tarvetta. Mikro-ohjain menee ohjelmoitavaan tilaan silloin, kun reset oikosulkupala asetetaan paikoilleen HWB oikosulkupalan ollessa asetettuna. (Atmel 2009, 246 – 250.)

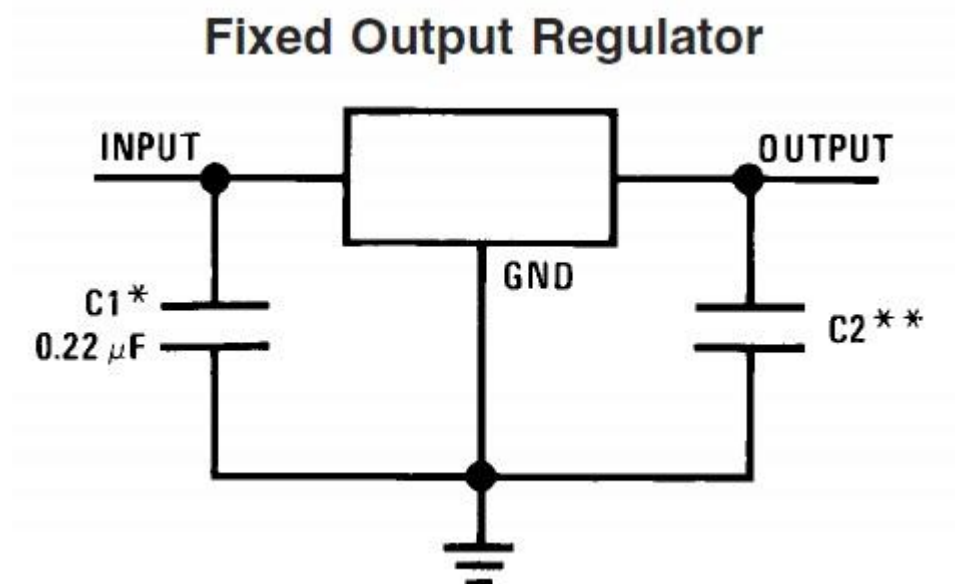


KUVIO 6. Tyypillinen USB-liitännästä käyttöjännitteen ottava Device-kytkentä (Atmel 2009)

4.3 Jännitteen regulointi

Koska tasavirtamoottori käyttää 9 voltin jännitettä ja kaikki muu elektroniikka 5 voltin käyttöjännitettä, oli käytettävä joko kahta erillistä virtalähdettä tai jänniteregulaattoria. Päädyin käyttämään jänniteregulaattoria, sillä se vaikutti järkevämältä ratkaisulta sovellukseeni. Regulaattorin valintaa tehdessä en vielä tiennyt moottorin käyttöjännitettä, mutta ajattelin sen olevan 9 ja 24 voltin välillä. Lisäksi tuli ottaa huomioon se että virtalähteenä sovelluksessani tulisin käyttämään pattereita, joiden jännite käyttöiän edetessä laskee, joten regulaattorin on pystyttävä sopeutumaan siihen.

Regulaattoriksi valitsin National semiconductorsin LM340 5 voltin jänniteregulaattorin. Valintaan vaikuttaviksi ominaisuuksiksi nousi 1 ampeerin jatkuvan virran kesto, sekä tulojännitteen 7.5 – 36 voltin haarukka. Kuviossa 7 on National Semiconductorin datalehdessä suosittelema kytkentä jännitteen regulointia varten. (National Semiconductor 2006.)

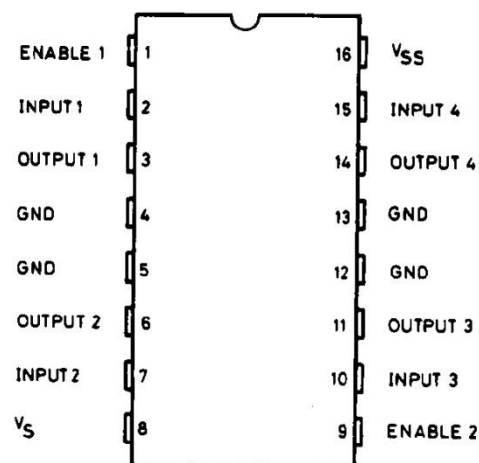


KUVIO 7. Tyypillinen kytkentä LM340 jännitteen regulointia varten (National Semiconductor 2006)

4.4 L293D-ohjainpiiri

Aluksi suunnitellessani laitetta en ajatellut tarvitsevani peruutusominaisuutta lainkaan. Työn edetessä huomasin kuitenkin, että sovelluksen pohjana käyttämäni vanhan rc-auton rungon renkaiden kääntymisakseli ei ollut riittävän suuri selviytymään jyrkemmissä mutkista. Tähän ongelmaan suunnittelin laitteeseen peruutusominaisuuden, joka toimii, kun auto huomaa olevansa joutumassa ulos linjalta. Peruutusta varten tarvitsisin H-sillan, jolla voidaan muuttaa virran kulkusuuntaa moottorin lävitse. Minulla oli aikaisemmasta elektroniikkaprojektista jäänyt ylimääräinen L293D-ohjainpiiri, joten tutkin, olisiko sitä mahdollista hyödyntää tähän tarkoitukseen.

L293D-ohjaimesta löytyy kaksi H-siltaa sekä diodit estämään sähkömoottorin käämeihin kertynyttä sähkömagneettista energiaa purkautumasta takaisin ohjainpiirille ja vahingoittamasta tätä. Ongelmaksi ilmeni, että L293D pystyy siirtämään 600 mA:n jatkuvan virran ja sietämään 1,2 ampeerin virtapiikkejä 100 mikrosekunnin ajan. Moottorini kuitenkin käytti noin 1 ampeeria jatkuvaa virtaa sekä tuotti noin 1,5 ampeerin virtapiikin käynnistyessään. Tutkittuani elektroniikka-aiheisia keskustelupalstoja selvisi, että on mahdollista kytkeä piiriin yksi moottori käyttäen molempia L293D:stä löytyviä kanavia, jolloin virrankesto tuplaantuu. Varmistin asian vielä datalehdestä ja totesin sen olevan käytännöllinen sekä kustannuksiltaan hyvä ratkaisu. (ST Microelectronics 2003.)



KUVIO 8. L293D-pinnijärjestys (ST Microelectronics 2003)

Seuraavaksi täytyi suunnitella miten syöttäisin ohjainsignaalit piirille. Kuviossa 8 on L293D-ohjainpiirin pinnijärjestys. Päädyin syöttämään PWM-signaalin D293D:n pinneihin 2 ja 3, suunnan valintasignaalin kytkin pinneihin 1 ja 4. Enable-pinnit kytkin myös mikro-ohjaimen lähtöihin, jotta pystyn niitä kontrolloimaan pysäyttämään moottorin virhetilan sattuessa. Taulukossa 1 on kuvattu totuustaulu pinnien kytkennästä. Tämänlaisen kytkennän myötä täytyi muistaa se, että suunnanvalintapinnin ollessa 1 moottori pyörii myötäpäivään PWM-signaalin ollessa 0, jolloin duty cycle on käänteinen normaaliin verrattuna. Suuntapinnin ollessa 0 moottori pyörii vastapäivään, PWM-signaalin ollessa 1 eli normaalin duty cyclen mukaan. Tämä seikka tuli ottaa huomioon ohjelmakoodia laadittaessa.

TAULUKKO 1. Totuustaulu L293D-ohjainpiirin toiminnasta

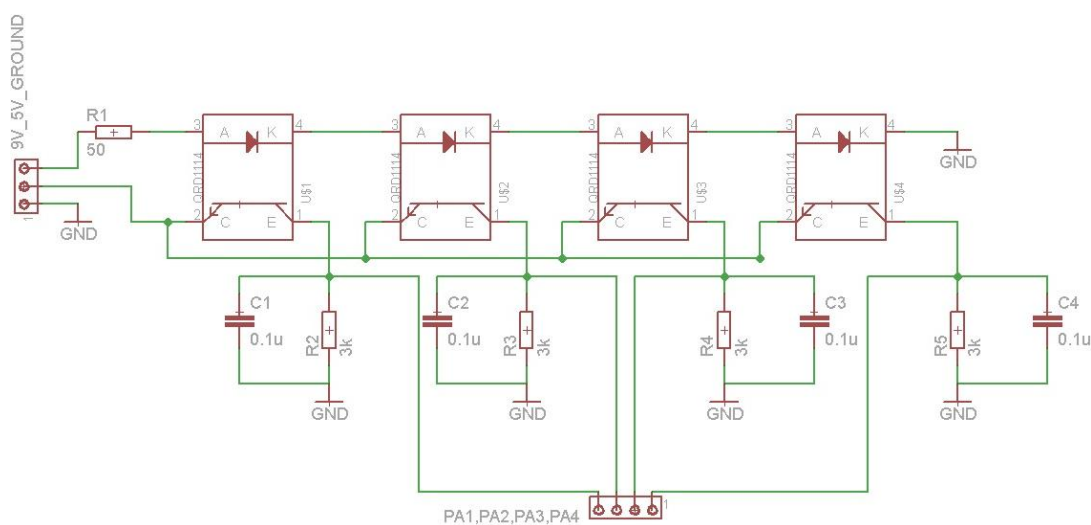
Pinnit	2 + 3 (PWM)	1 + 4 (Suunta)	Enable
Pysähdys	0	0	1
Myötäpäivään	0	1	1
Vastapäivään	1	0	1
Pysähdys	1	1	1
Pysähdys	1/0	1/0	0

4.5 IR sensori QRD1114

Linjan seuranta varten tarvitsin infrapuna-sensoreita sekä niille valoa tuottamaan infrapuna-ledejä. Tutkin erinäisiä vaihtoehtoja, mutta päädyin lopulta Fairchild Semiconductorin valmistamaan QRD1114-sensorilediyhdistelmään, jossa on kätevästi IR-ledi sekä sensori samassa pakkauksessa. QRD1114:stä löytyy myös valmiina päivänvalosuodatin, joka suodattaa pois mahdollisia virheitä tuottavia säteitä ympäristöstä. Normaalisti kytkettynä sensori palauttaa mikrokontrollerille analogisen arvon heijastuksen suuruudesta. Suunnittelinkin aluksi jokaiselta sensorilta tulevan signaalin saapuvan mikrokontrollerin A/D muunnokseen pystyviin pinneihin PA1 – PA4. Sovellustani varten tarvitsen kuitenkin vain tiedon siitä, onko sensorin alla oleva materiaali mustaa vai valkoista. Päädyin muuttamaan sensorilta tulevan analogiasignaalin digitaaliseksi, lisäämällä alavetovastusten

kanssa rinnan $0.1 \mu\text{F}$:n kondensaattorit. Tällöin mikro-ohjainta ei tarvitse kuormittaa A/D-muunnoksella ja pystyn tekemään ohjelmakoodista kevyemmän. Kuviossa 9 näkyy sensoreiden kytkentäkaavio. Alasvetovastusten R2 – R5 avulla on mahdollista muuttaa sensoreiden herkkyyttä. Vastuksen kasvaessa sensorin herkkyys kasvaa ja sensori havaitsee kohteet jo kauempaa, mutta lukee myös useammin virheellisesti mustan valkoiseksi.

Vastuksia valitessa jouduin ensin kokeilemaan säätövastuksen avulla mikä olisi toimivin vastuksen suuruus sovellukseeni. Päädyin lopulta käyttämään kiinteitä $5 \text{ k}\Omega$ vastuksia. Vastuksen suuruuteen vaikuttaa sensoreiden etäisyys maasta sekä seurattavan viivan heijastuskyky. Kuvioista 8 voi myös huomata, että ledeille tulee 9 voltin käyttöjännite, näin on tehty virransäästösyistä, sillä sarjassa olevat ledit tarvitsevat vain yhden virtaa säätelevän vastuksen. Sarjaan kytkettäessä ledit kuitenkin tarvitsevat suuremman jännitteen, sillä ledien myötäjännitearvot lasketaan yhteen. QRD1114-sensoreiden tapauksessa myötäjännite nousi $6,8 \text{ V}$ jolloin 5 V voltin käyttöjännite ei riitä sytyttämään niitä. Sarjaankytketyille ledeille voidaan laskea virransäätövastus vähentämällä yhteenlaskettu myötäjännitearvo käyttöjännitteestä, ja kertomalla tulos jatkuvan virran kestolla. Tässä tapauksessa $(9 \text{ V} - 6,8 \text{ V}) / 0,05 \text{ mA} = 44 \Omega$, vastus kannattaa aina mitoittaa yläkanttiin, sillä ledit voivat rikkoontua herkästi. (Fairchild semiconductor 2005.)



KUVIO 9. QRD1114-sensoreiden kytkentäkaavio

4.6 Prototyypin rakentaminen

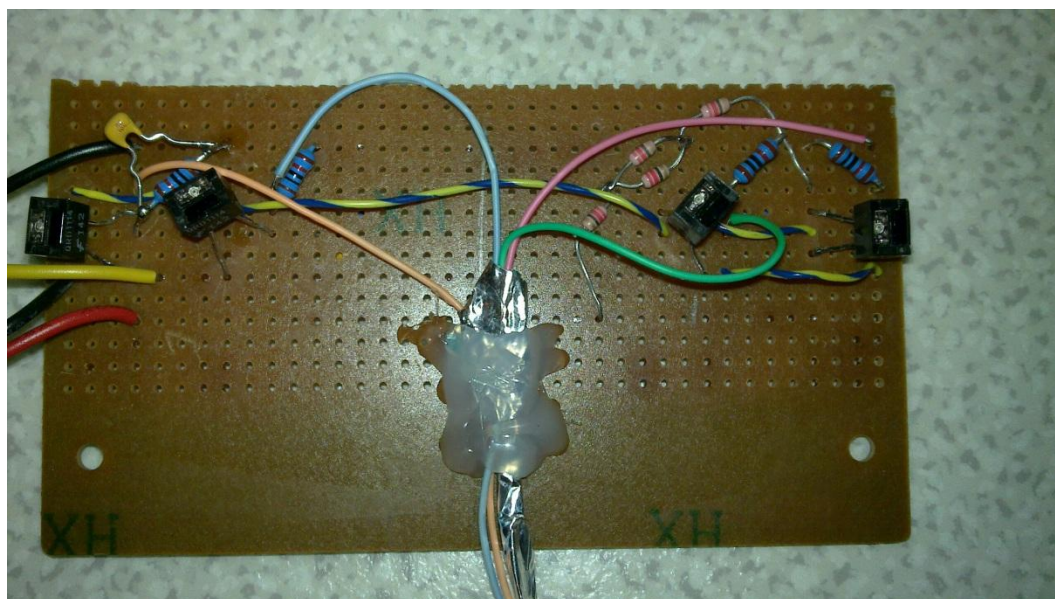
Aloitin työn suunnittelun luomalla piirikaavion Cadsoftin Eagle (Easily Applicable Graphical Layout Editor) piirilevynsuunnitteluohjelmistolla. Piirikaavion suunnittelun tarkoituksena oli helpottaa projektin hahmoittamista kokonaisuutena, vaikkei minun ollutkaan tarkoitus tuottaa sovellukselleni piirilevyä, sillä sovelluksen tarkoituksena oli moottorin ohjauksen demonstrointi eikä sovelluksesta ole oikeastaan käytännön hyötyä. Suunnitellut piirikaaviot löytyvät opinnäytetyön liitteestä 1.

Suunnitelman ollessa pääpiirteittäin valmis tilasin tarvittavat osat yhdysvaltalaisesta elektroniikkaliikkeestä nimeltä DigiKey. Hinnat tulivat veroineen sekä postituskuluineen silti edullisemmaksi kuin suomalaisten liikkeiden hinnat. Projektin kannalta olennaisten osien lisäksi tilasin reilusti erikokoisia vastuksia, kondensattoreita sekä diodeja, sillä olen huomannut, että prototyyppinä rakentaessa tulee peruskomponenteille aina käyttökohteita.

Päädyin käyttämään työn rakentamisen pohjana kuparitonta protoalustaa valmiilla 1,07 mm rei'ityksellä. Protoalusta toimi tarkoitukseen hyvin, mutta yhtenäiset maa- sekä käyttöjännitepinnat olisivat helpottaneet juottamista.

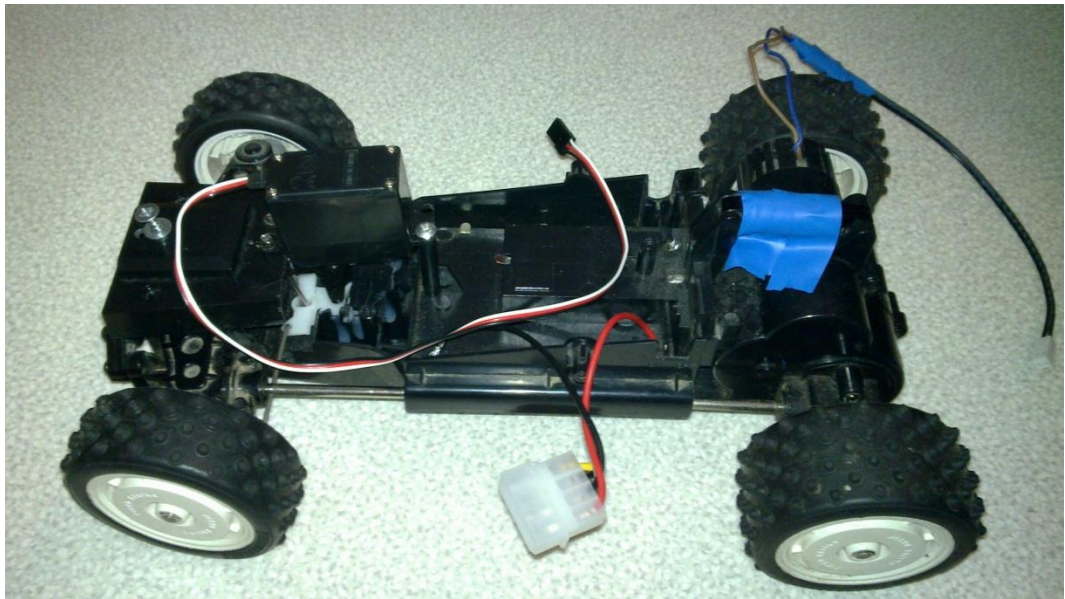
Kolvauksen aloitin rakentamalla USB-liitännän, sen tarvitsemat oheislaitteet kuten 8 MHz ulkoisen oskillaattorin sekä OC0A-kanavan ulostulopinnin. Näiden jälkeen kokeilin USB-liitännän sekä PWM-kanavan toimintaa. USB-liitäntä toimi moitteetta, mikä ei kuitenkaan tullut yllätyksenä, sillä olin käyttänyt vastaavaa kytkentää jo aikaisemmin. PWM-kanavan toimintaa havainnollistin valodiodia himmentämällä ja kirkastamalla koekytkentäalustalla. LED ei kuitenkaan toiminut suunnitellusti, vaan vilkkui tiuhaan tahtiin. Tähän onglemaan etsin pitkään ratkaisua ohjelmakoodistani, mutta huomasin lopulta ongelman korjaantuvan, kun mikro-ohjaimen resetoit muutaman kerran. Lisäsin vielä reset sekä hardwareboot-painikkeiden ylösvetovastusten rinnalle 220 nF:n kondensattorit poistamaan mahdollista näppäinvärähtelyä. En kondensattoreiden lisäämisen jälkeen enää havainnut tätä ongelmaa.

Seuraavaksi oli vuorossa robotin alle sijoitettavien QRD1114-sensoreiden kytkeminen, kytkin sensorit omalle protoalustalleen. Tässä alustassa oli muutamia valmiiksi vedettyjä kuparivetoja, joita käytinkin maa- sekä käyttöjännitetasojen juottamiseen. Sensorialusta toimi suunnitellusti ja siirryin eteenpäin. Vasta myöhemmin kokeiltaessa robottia toiminnassa huomasin, että mustana käyttämäni materiaali heijasti liian voimakkaasti infrapunavaloa takaisin sensorille, ja se antoi virheellisiä arvoja. Tämä korjaantui tiputtamalla alasvetovastusten arvoa puolella kytkemällä rinnalle toinen $10\text{ k}\Omega$ vastus. Tällä tavoin saatiin sensoreiden herkkyys pienemmäksi eikä virhearvoja enää tullut. Toinen jälkeinpäin tehty muokkaus oli $0.1\text{ }\mu\text{F}$ kondensaattoreiden lisäys vastusten kanssa rinnan, jotta sensoreiden signaali saadaan digitaaliseksi. Kuvioista 10 näkyy $22\text{ }\Omega$ vastusten verkko, jolla korvasin $55\text{ }\Omega$ valodiodien virranrajoitusvastuksen, sillä sopivaa vastusta ei ollut saatavilla. Kuvioista 10 näkyy myös jälkeinpäin lisätty kondensaattori rinnan alasvetovastuksen kanssa.



KUVIO 10. Kuva valmiista QRD1114-sensoreiden alustasta

Seuraava ongelma ilmeni robotin runkona käytetyn RC-auton kääntämismekanismissä. Tiesin entuudestaan että sen kääntöakseli oli pieni, mutta oletin mekanismina olevan normaalisti RC-autoissa käytetty servomoottori. Tämä osoittautui vääräksi olettamukseksi, sillä kääntymismekanismina oli käytetty sähkömagneettia ja palautukseen jousia. Sähkömagneettiin johtamalla virtaa magneetti veti puoleensa akseliin kiinnitettyjä magneetteja riippuen virran kulkusuunnasta. Virran ollessa poikki jouset palauttivat magneetit keskiasentoon. Poistin tämän magneettijärjestelmän ja asensin tilalle servomoottorin. Tämä ei kuitenkaan ollut aivan yksinkertainen lisäys, sillä mekanismit olivat täysin erilaiset. Kuumaliiman sekä muutamien ruuvien avulla sain servomoottorin kuitenkin toimimaan halutulla tavalla. Kuviossa 11 on RC-auton runko, rungosta on poistettu vanha ohjauselektronikka ja sen tilalle on lisätty servomoottori.



KUVIO 11. RC-auton runko, johon on lisätty servomoottori

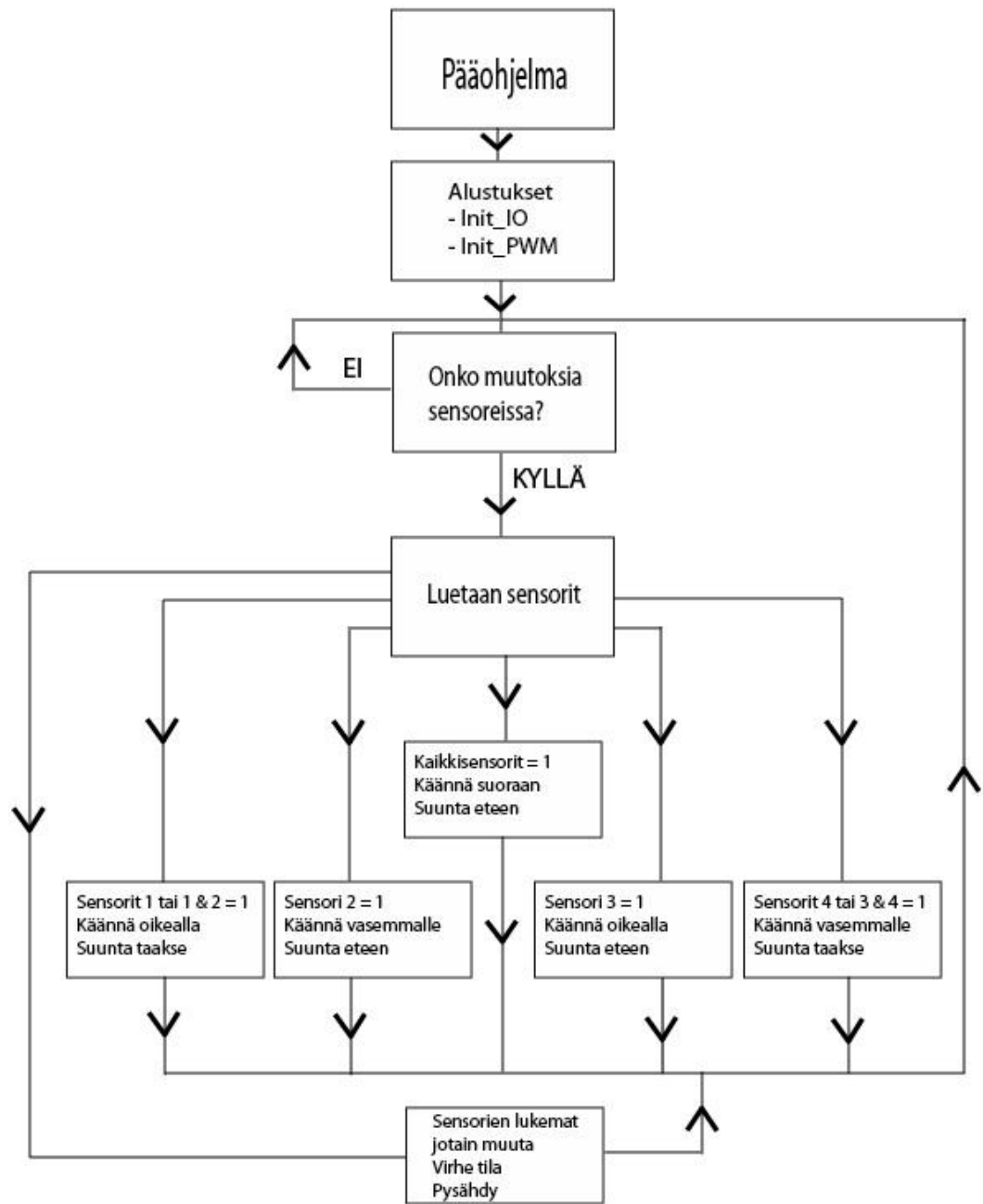
5 OHJELMAKOODIN SUUNNITTELU

Ohjelmakoodin suunnittelun aloitin tekemällä vuokaavion ohjelman toiminnasta (kuvio 12). Kun ohjelman toimintaperiaate oli pääpiirteittäin valmis, aloin ohjelmakoodin luomisen. Ohjelmoinnin suoritin C-ohjelmointikielellä käyttäen Atmelin ilmaista AVR Studio 4 -kehitysympäristöä. AVR Studio sisältää myös mahdollisuuden simuloida ohjelmakoodin toimintaa. Simulointi on hyödyllinen apuväline koodin kirjoituksessa, koska sillä pystyy nopeasti havaitsemaan ongelmakohdat sovelluksessa. Huomasin kuitenkin työtä tehdessäni, ettei simulaattoriin kannata luottaa sokeasti, vaan kannattaa aina mahdollisuuksien mukaan kokeilla koodia myös suoraan kohdelaitteessa. Esimerkiksi korkeampien PWM-toimintatapojen kohdalla simulaattori toimi väärin, mutta laitteessa koodi toimi moitteetta. Hukkasinkin useita tunteja etsiessäni virhettä koodistani, vaikka ongelma oli simulaattorissa.

Pääohjelman alussa ohjelma kutsuu kahta funktiota. Ensimmäisessä funktiossa `IO_init()` alustetaan käytettävät portit tarpeen mukaan tuloiksi ja lähdöiksi. Jälkimmäinen funktio `PWM_Init()` ottaa PWM-toiminnot käyttöön sekä antaa aloitusarvot vertailurekistereille.

Alustusten jälkeen siirrytään niin kutsuttuun ikuiseen silmukkaan. Silmukassa tarkastellaan IR-sensoreilta tulevia signaaleita, ja jos signaaleissa tulee muutoksia, ohjelma tekee ennalta määritetyt toimenpiteet. Toimenpiteisiin kuuluu L293D-ohjainpiirille lähetettävät ohjainsignaalit sekä muutokset vertailurekistereihin. Näiden jälkeen ohjelma palaa tarkkailemaan muutoksia IR-sensoreilta tulevissa signaaleissa.

Toteutin sensoreiden tarkkailun ikuisessa silmukassa IF-lausekkeiden ketjulla. Tarkoituksena oli hyödyntää myös keskeytyksiä ohjelmakoodissa, jolloin prosessori ei kuormittuisi niin paljoa jatkuvan sensoreiden tarkkailun myötä, vaan se voisi tarkastaa sensorit vain silloin, kun niissä on muutoksia. Ajan loppumisen vuoksi en kuitenkaan kerinnyt perehtymään keskeytyksiin syvällisemmin ja se jäi sovelluksesta tässä vaiheessa puuttumaan.



KUVIO 12. Vuokaavio ohjelmiston toiminnasta

6 TYÖN ANALYSOINTI

Työni lähtökohta oli PWM-tekniikan demonstrointi, jota varten suunnittelin ja rakensin linjanseurantarobotin. Onnistuin myös vaikeuksista huolimatta lisäämään servojärjestelmällä ohjatun tasavirtamoottorin sovellukseen. Onnistuin työn toteutuksessa mielestäni hyvin, vaikka prototyyppi ei ulkonäöltään ole kovin hiottu. Opinnäytetyö oli monipuolinen ja mielestäni riittävän haastava.

Ongelmia esiintyi työn suunnittelun ja toteuttamisen aikana, mutta pystyin ratkaisemaan ne osin itsenäisesti ja osin Internetin keskustelupalstoja selaten. Huomasin, että sellaista ongelmaa harvoin esiintyy, mihin joku muu ei olisi jossain määrin aikaisemmin törmännyt, ja usein muiden ratkaisuja soveltamalla pystyin selvittämään eteen tulleita ongelmia.

Työn suunnittelun ja toteutuksen aikana taitoni varsinkin analogiaelektroniikan osalta kehittyivät. Opiskelun aikana teorialuennoilla esiteltyjä asioita palautui mieleen ja niiden tarkoituksen ymmärsi oikeastaan vasta hyödyntämällä niitä käytännössä. Ohjelmointi on mielestäni vahvin osa-alueeni, enkä sen osalta oppinut uutta projektin aikana. Tämä johtui osaltaan ajan puutteesta, työn suunnittelu sekä rakennusvaihe veivät niin kauan, että ohjelmoinnille ei jäänyt paljoa aikaa ja jouduin tekemään melko yksinkertaisen ohjelmiston. PWM-signaaleihin tutustuin työn puolesta kattavasti ja uskoisin pystyväni saamaan PWM-ominaisuudet käyttöön kaikista nykyaikaisista mikro-ohjaimista. Olenkin jo tutustunut NXP:n Cortex M3-sarjan mikro-ohjaimiin ja siihen miten niillä tuotetaan PWM- ohjaussignaalia.

Tarkoituksena on muokata sovelluksen ohjelmistoa vielä tulevaisuudessa paremmaksi sekä virhesietoisemmaksi sekä toteuttaa sensoreiden lukeminen keskeytyksia hyödyntämällä. Tällöin mikro-ohjain ei kuormitu turhaan sensoreiden jatkuvan tarkkailun vuoksi.

Useat datalehdet tulivat työn suunnittelun sekä toteutuksen aikana hyvin tutuiksi, ja projektin edetessä opin seulomaan tärkeän tiedon epäselvistäkin datalehdistä. Varsinkin AT90USB1287 datalehti tuli työtä tehdessä tutuksi PWM-kanavien käyttöönoton yhteydessä. Atmelin datalehdet ovat datalehtien osalta parhaimmista ja melko selkeästi toteutettuja, ja se säästi minulta paljon aikaa.

Vaikka sovellus toimi odotusten mukaisesti, olisi siihen luotettavan toiminnan takaamiseksi hyvä tehdä muutamia muutoksia. Ensimmäkin seurattavassa linjassa voi helposti olla jyrkkiä mutkia tai silmukoita, eikä nykyinen sovellus pysty kääntymään kovin jyrkästi. Tätä korjaamaan kehitin sovellukseen peruutusmekanismi, mutta peruuttaminen hidastaa matkantekoa. Rakentaisin seuraavaksi version, jossa robotissa olisi kaksi pyörää ja molemmilla pyörillä olisi omat moottorit, joiden avulla pystyisi ohjaamaan robottia pyörimissuuntaa tai nopeutta muuttamalla. Tällä tavoin robotti pystyisi kääntymään jopa 360° paikalla ollessaan.

Projektia voisi myös laajentaa hieman ja suunnitella robotille erilaisia käyttötarkeitä. Esimerkiksi pienillä muutoksilla robotista voisi muokata esteitä väistelevän siivousrobotin.

LÄHTEET

Atmel. 2009. 8-bit Microcontroller with 64/128K bytes of ISP Flash and USB Controller [viitattu 8.4.2011]. Saatavissa:

http://www.atmel.com/dyn/resources/prod_documents/doc7593.pdf

PWM PIC: Pulse Width Modulation. 2011. PWM for the PIC Microcontroller (or any microcontroller) [viitattu 3.4.2011]. Saatavissa: [http://www.best-](http://www.best-microcontroller-projects.com/pwm-pic.html)

[microcontroller-projects.com/pwm-pic.html](http://www.best-microcontroller-projects.com/pwm-pic.html)

Bräunl, T. 2008. Embeded robotics. 3. painos. Berlin, Heidelberg: Springer-Verlag.

Eleinmec.com. 2007. Pulse Width modulation [viitattu 15.3.2011]. Saatavissa:

<http://www.eleinmec.com/article.asp?28>

Extreme electronics. 2007. Servo Motor control by using AVR ATmega32 Micro controller [viitattu 23.3.2011]. Saatavissa: [http://extremeelectronics.co.in/avr-](http://extremeelectronics.co.in/avr-tutorials/servo-motor-control-by-using-avr-atmega32-microcontroller/)

[tutorials/servo-motor-control-by-using-avr-atmega32-microcontroller/](http://extremeelectronics.co.in/avr-tutorials/servo-motor-control-by-using-avr-atmega32-microcontroller/)

Fairchild Semiconductor. 2008. QRD1113, QRD1114 Reflective Objective Sensor [viitattu 8.4.2011]. Saatavissa:

<http://www.fairchildsemi.com/ds/QR/QRD1114.pdf>

Johnson, J. & Kördel, L. 2003. Servotekniikka. Iisalmi: IS-PRINT.

Lappeenranta university of technology 2011. Sähkömoottori [viitattu 5.4.2011].

Saatavissa:

http://www.lut.fi/fi/technology/lutenergy/electrical_engineering/articles/electrical_motor/Sivut/Default.aspx

National Semiconductor. 2006. LM340/LM78XX Series 3-Terminal Positive Regulators [viitattu 8.4.2011]. Saatavissa:

<http://www.national.com/ds/LM/LM340.pdf>

Netrino.com. 2001. Introduction to pulse width modulation (PWM) [viitattu 15.3.2011]. Saatavissa: <http://www.netrino.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>

Parallax. 2010. Parallax Standard Servo [viitattu 8.4.2011]. Saatavissa: <http://www.parallax.com/Portals/0/Downloads/docs/prod/motors/900-00005StdServo-v2.1.pdf>

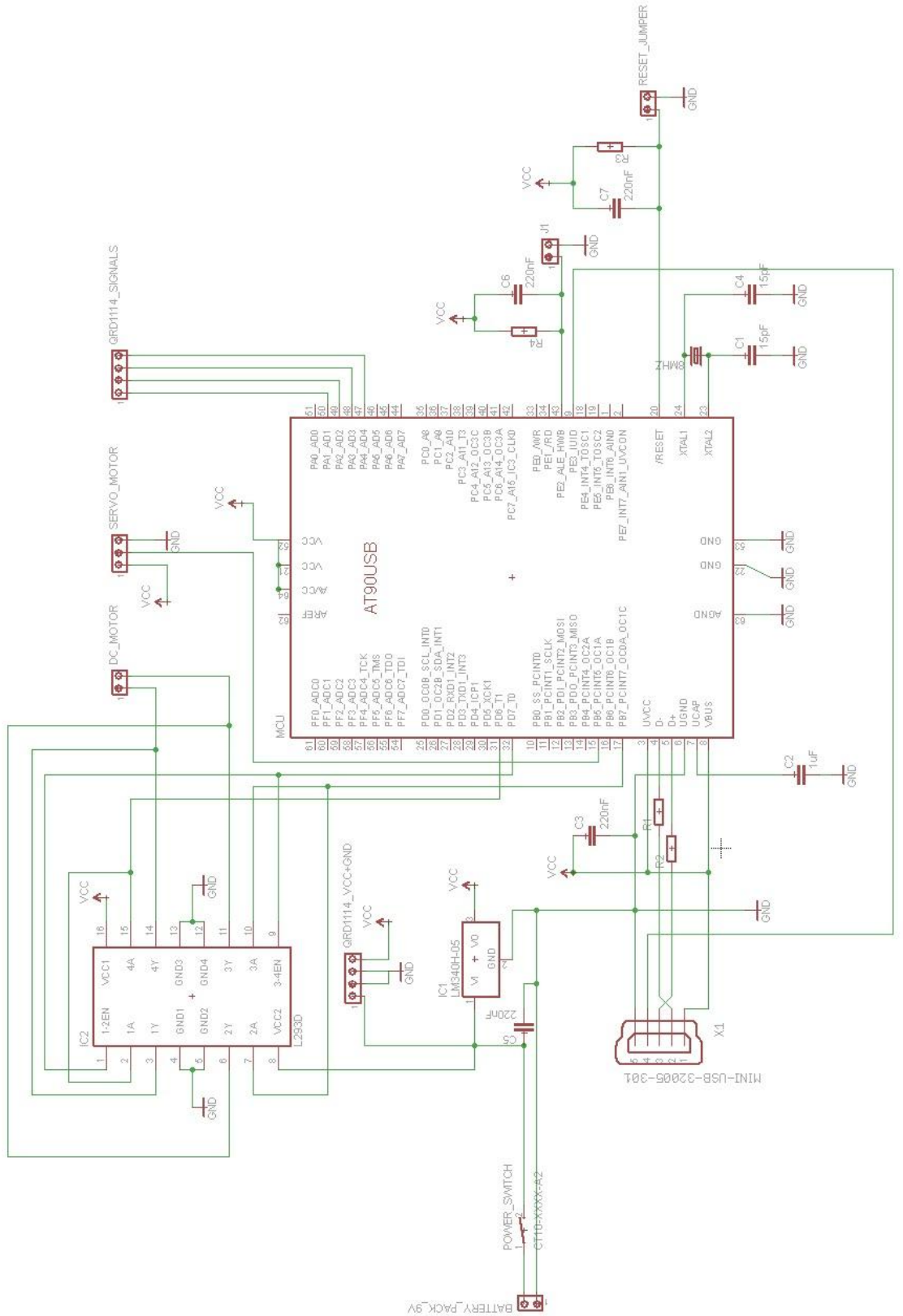
ST Microelectronics. 2003. Push-Pull four channel driver with diodes [viitattu 8.4.2011]. Saatavissa: <http://www.st.com/stonline/books/pdf/docs/1330.pdf>

LIITTEET

Liite 1: Ohjainpiirin kytkentäkaavio

Liite 2: Ohjelmakoodi

The schematic illustrates the electrical connections for a custom PCB. The central component is the AT90USB microcontroller, which is interfaced with a DC motor, a servo motor, and a motor driver (L293D). The board includes a power switch, a battery pack, a reset button, and various passive components like capacitors and resistors. The schematic shows the internal connections of the microcontroller, the motor driver, and the power management components.



Liite 2. Ohjelmakoodi

```
#include <avr/io.h>
#define F_CPU 1000000UL
#include <util/delay.h>
#include <avr\interrupt.h>

void Init_IO()
{
    DDRA&=(0<<PA1)|(0<<PA2)|(0<<PA3)|(0<<PA4);
    DDRD|=(1<<PD7)|(1<<PD6);
    DDRB|=(1<<PB7)|(1<<PB5);
    PORTD&=(0<<PD6);
    PORTD|=(1<<PD7);
}

void Init_PWM()
{
    TCCR0A|=(1<<WGM00)|(1<<WGM01)|(1<<COM0A1)|(1<<COM
0A0);
    TCCR0B|=(1<<CS00)|(1<<CS01);

    TCCR1A|=(1<<COM1A1)|(1<<WGM11);
    TCCR1B|=(1<<WGM12)|(1<<WGM13)|(1<<CS10)|(1<<CS11);

    OCR0A=0x80;
    ICR1H=0x01;
    ICR1L=0x38;
    OCR1A=0x17;
}

void Wait()
{
    _delay_loop_2(0);
}
```

Liite 2. (jatkuu)

```
void main()
```

```
{
```

```
    Init_IO();
```

```
    Init_PWM();
```

```
    uint8_t p = 0;
```

```
    while(1)
```

```
    {
```

```
        if(PINA & 0x10 && PINA & 0x08 && PINA & 0x04
```

```
        && PINA & 0x02)
```

```
        {
```

```
            if(p != 1)
```

```
            {
```

```
                p = 1;
```

```
                PORTD&=(0<<PD7);
```

```
                OCR1A=0x1A;
```

```
                PORTD=0x80;
```

```
                OCR0A=0x60;
```

```
            }
```

```
        }
```

```
        else if(~PINA & 0x02)
```

```
        {
```

```
            if(p != 2)
```

```
            {
```

```
                p = 2;
```

```
                PORTD&=(0<<PD7);
```

```
                OCR1A=0x1E;
```

```
                OCR0A=0xA0;
```

```
                while(~PINA & 0x10)
```

```
                {
```

```
                    PORTD=0xC0;
```

```
                }
```

```
                Wait();
```

Liite 2. (jatkuu)

```
}
```

```
}
```

```
else if(~PINA & 0x04)
```

```
{
```

```
    if(p != 3)
```

```
    {
```

```
        p = 3;
```

```
        PORTD&=(0<<PD7);
```

```
        OCR0A=0xA0;
```

```
        OCR1A=0x16;
```

```
        while(~PINA & 0x08)
```

```
        {
```

```
            PORTD=0xC0;
```

```
        }
```

```
        Wait();
```

```
    }
```

```
}
```

```
else if(~PINA & 0x08)
```

```
{
```

```
    if(p != 4)
```

```
    {
```

```
        p = 4;
```

```
        PORTD&=(0<<PD7);
```

```
        OCR1A=0x1E;
```

```
        PORTD=0x80;
```

```
        OCR0A=0x60;
```

```
    }
```

```
}
```

```
else if(~PINA & 0x10)
```

```
{
```

```
    if(p != 5)
```

Liite 2. (jatkuu)

```

    {
    p = 5;
    PORTD&=(0<<PD7);
    OCR1A=0x16;
    PORTD=0x80;
    OCR0A=0x60;

    }
else
{
    if(p != 6)
    {
    p = 6;
    PORTD&=(0<<PD7);
    OCR1A=0x1A;
    PORTD=0xC0;
    OCR0A=0x90;
    }}}}

```